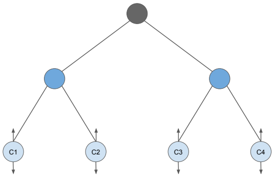
# Introduction to Machine Learning

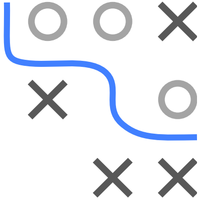
# Gradient Boosting with Trees 2




**Learning goals**

- Loss optimal terminal coefficients
- GB with trees for multiclass problems

# ADAPTING TERMINAL COEFFICIENTS



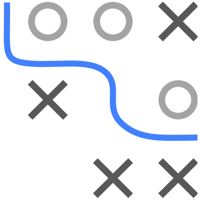
- Tree as additive model: $b(\mathbf{x}) = \sum_{t=1}^{T} c_t \mathbb{1}_{\{\mathbf{x} \in R_t\}}$,
- $R_t$ are the terminal regions; $c_t$ are terminal constants

The GB model is still additive in the regions:

$$
\begin{aligned}
f^{[m]}(\mathbf{x}) &= f^{[m-1]}(\mathbf{x}) + \alpha^{[m]} b^{[m]}(\mathbf{x}) \\
&= f^{[m-1]}(\mathbf{x}) + \alpha^{[m]} \sum_{t=1}^{T^{[m]}} c_t^{[m]} \mathbb{1}_{\{\mathbf{x} \in R_t^{[m]}\}} \\
&= f^{[m-1]}(\mathbf{x}) + \sum_{t=1}^{T^{[m]}} \tilde{c}_t^{[m]} \mathbb{1}_{\{\mathbf{x} \in R_t^{[m]}\}}.
\end{aligned}
$$

With $\tilde{c}_t^{[m]} = \alpha^{[m]} \cdot c_t^{[m]}$ in the case that $\alpha^{[m]}$ is a constant learning rate

# ADAPTING TERMINAL COEFFICIENTS

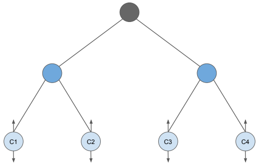


After the tree has been fitted against the PRs, we can adapt terminal constants in a second step to become more loss optimal.

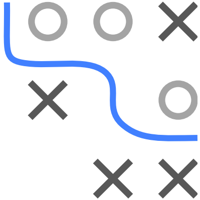
$$f^{[m]}(\mathbf{x}) = f^{[m-1]}(\mathbf{x}) + \sum_{t=1}^{T^{[m]}} \tilde{c}_t^{[m]} \mathbb{1}_{\{\mathbf{x} \in R_t^{[m]}\}}.$$

We can determine/change all $\tilde{c}_t^{[m]}$ individually and directly $L$-optimally:

$$\tilde{c}_t^{[m]} = \arg\min_c \sum_{\mathbf{x}^{(i)} \in R_t^{[m]}} L(y^{(i)}, f^{[m-1]}(\mathbf{x}^{(i)}) + c).$$



# ADAPTING TERMINAL COEFFICIENTS



An alternative approach ist to directly fit a loss-optimal tree. Risk for data in a node:

$$\mathcal{R}(\mathcal{N}') = \sum_{i \in \mathcal{N}'} L(y^{(i)}, f^{[m-1]}(\mathbf{x}^{(i)}) + c)$$

with $\mathcal{N}'$ being the index set of a specific (left or right) node after splitting and $c$ the constant of the node.

$c$ can be found by line search or analytically for some losses.

# GB MULTICLASS WITH TREES

- From Friedman, J. H. - Greedy Function Approximation: A Gradient Boosting Machine (1999)
- We again model one discriminant function per class.
- Determining the tree structure works just like before.
- In the estimation of the $c$ values, i.e., the heights of the terminal regions, however, all models depend on each other because of the definition of $L$. Optimizing this is more difficult, so we will skip some details and present the main idea and results.

# GB MULTICLASS WITH TREES

- There is no closed-form solution for finding the optimal $\hat{c}_{tk}^{[m]}$ values. Additionally, the regions corresponding to the different class trees overlap, so that the solution does not reduce to a separate calculation within each region of each tree.
- Hence, we approximate the solution with a single Newton-Raphson step, using a diagonal approximation to the Hessian (we leave out the details here).
- This decomposes the problem into a separate calculation for each terminal node of each tree.
- The result is

$$\hat{c}_{tk}^{[m]} = \frac{g-1}{g} \frac{\sum_{\mathbf{x}^{(i)} \in R_{tk}^{[m]}} \tilde{r}_k^{[m](i)}}{\sum_{\mathbf{x}^{(i)} \in R_{tk}^{[m]}} \left|\tilde{r}_k^{[m](i)}\right| \left(1 - \left|\tilde{r}_k^{[m](i)}\right|\right)}.$$

# GB MULTICLASS WITH TREES

**Algorithm** Gradient Boosting for $g$-class Classification.

Initialize $f_k^{[0]}(\mathbf{x}) = 0,\ k = 1, \ldots, g$

**for** $m = 1 \to M$ **do**

  Set $\pi_k(\mathbf{x}) = \frac{\exp(f_k^{[m]}(\mathbf{x}))}{\sum_j \exp(f_j^{[m]}(\mathbf{x}))}, k = 1, \ldots, g$

  **for** $k = 1 \to g$ **do**

    For all $i$: Compute $\tilde{r}_k^{[m](i)} = \mathbb{1}_{\{y^{(i)}=k\}} - \pi_k(\mathbf{x}^{(i)})$

    Fit regr. tree to the $\tilde{r}_k^{[m](i)}$ giving terminal regions $R_{tk}^{[m]}$

    Compute

$$\hat{c}_{tk}^{[m]} = \frac{g-1}{g} \frac{\sum_{\mathbf{x}^{(i)} \in R_{tk}^{[m]}} \tilde{r}_k^{[m](i)}}{\sum_{\mathbf{x}^{(i)} \in R_{tk}^{[m]}} \left|\tilde{r}_k^{[m](i)}\right| \left(1 - \left|\tilde{r}_k^{[m](i)}\right|\right)}$$

    Update $\hat{f}_k^{[m]}(\mathbf{x}) = \hat{f}_k^{[m-1]}(\mathbf{x}) + \sum_t \hat{c}_{tk}^{[m]} \mathbb{1}_{\{\mathbf{x} \in R_{tk}^{[m]}\}}$

  **end for**

**end for**

Output $\hat{f}_1^{[M]}, \ldots, \hat{f}_g^{[M]}$